International Journal of Computer Networking,
Wireless and Mobile Communications (IJCNWMC)
ISSN (P): 2250-1568; ISSN (E): 2278-9448
Vol. 9, Issue 1, Jun 2019, 5-16

EFFICIENT TRUST MANAGEMENT TECHNIQUE USING NEURAL NETWORK IN CLOUD COMPUTING

P.NEELAKANTESWARA[1] & P. SURYANARAYANA BABU[2]

[1]*Research Scholar, Rayalaseema University, Kurnool, Andhra Pradesh, India*

[2]*Principal, VijayaDurga Degree College, Kurnool, Andhra Pradesh, India*

ABSTRACT

Cloud computing turned into a standout amongst the most imperative stage for the cloud suppliers so as to introduce the cloud suppliers in a virtual way through the internet. Cloud customers throughout the world exchange their information with a high scope of computing resources, optional level of information stockpiling with high bandwidth. Subsequently, the progressing interest for the versatile resources is widely expanding between the cloud clients. Along these lines, single cloud server couldn't able to distinguish and unite with a high scope of ability to the application among runtime. Subsequently, the specialists are in need to fabricate virtual environment for interfacing the numerous cloud servers accordingly leads the researchers to the collaborative Cloud computing (CCC). This paper uses an effective resource sharing platform called Harmony and also uses the Neural Networks (NN) for suitable resource selection. Further, the trust management is implemented and optimal time period for resource selection is enhanced.

KEYWORDS: Cloud Computing, Task Scheduling, Load Balancing, Cost, Time & Qos

Received: Nov 17, 2018; **Accepted:** Dec 07, 2018; **Published:** Jan 31, 2019; **Paper Id.:** IJCNWMCJUN20192

Original Article

INTRODUCTION

In Cloud Computing [1] versatile assets are provisioned powerfully as an administration over the web so as to guarantee bunches of money related advantages' to be scattered among its adopters. Distinctive layers are sketched out in view of the sort of administrations gave by the Cloud. Moving from base to top, base layer contains fundamental equipment assets like Memory, Storage Servers. Thus it is signified as Infrastructure-as-a-Service (IaaS). The recognized case of IaaSis Amazon Simple Storage Service (S3) and Amazon Elastic Compute Cloud (EC2). The layer above IaaS is Platform-as-a-Service (PaaS) which primarily underpins arrangement and element scaling of Python and Java-based applications. One such a case of PaaS is Google App Engine. On top of PaaS, a layer that offers its clients with the capacity to utilize their applications alluded to as Software-as-a-Service (SaaS). SaaS underpins getting to client's applications through a program without the learning of Hardware or Software to be introduced. This methodology has been turned out to be an all-around acknowledged and trusted administration. Web and Browser are the two parts required to get to these Cloud administrations. IaaS applications access requires more web transfer speed whereas web program might be adequate with sensible web data transmission is adequate to get to SaaS and PaaS applications. "Cloud" was a code word for everything that was past the server farm or out on the system. There are a few meanings of a cloud accepted by various classes of cloud clients. It is generally depicted as programming as an administration, where clients can get to a product application on the web, as in Salesforce.com, Google Apps and Zoho. It is likewise portrayed as base as an administration, where a client does not claim foundation but rather and rents it after some time on a server and gets to through a site, for example,

Amazon Elastic Compute Cloud (EC2). Another type of a Cloud is Platform as an administration in which certain devices are made accessible to manufacture programming that keeps running in the host cloud. Fundamentally a cloud is worked over some of the server farms, which mirrors the Web's setting for approximately coupled frameworks (i.e.two frameworks don't think about each other), and gives the capacity to have virtualized remote servers through standard Web administrations to have substantial registering power. Cloud worldview likewise serves as a plan of action separated from innovation. Through the plan of action, the cloud makes another type of processing broadly accessible at lower costs that would have been viewed as unimaginable. Distributed computing can be additionally utilized for dispatching client errands or employments to the accessible framework asset like stockpiling and programming.

In distributed computing, scheduling assumes significant part to dispatch client undertakings and subsequently, it reflects as another example of business figuring. The fundamental system of Berger model in distributed computing is to dispatch the registering errands to asset pooling which is constituted by enormous PCs. It empowers an assortment of utilizations to pick up figuring force, stockpiling and an assortment of programming administrations as per their needs. The ancestors have actualized the calculations of occupation scheduling taking into account Berger Model in distributed computing keeping in mind the end goal to have the capacity to delineate hypothesis of distributive equity in Berger Model (BaominXu et al 2011) to asset allotment model in distributed computing. It is expected to bear on the undertaking characterization, reasonableness capacity meaning of client assignments, the errand and asset parameterization, the assignment, asset mapping, and so on. Taking into account the possibility of Berger model, two-decency imperatives of occupation scheduling are set up in distributed computing. In this, the client assignments are ordered taking into account Quality of Service parameters like data transmission, memory, CPU use, and size. The arranged undertakings are given to fuzzifier, neural system lastly defuzzifier. The model info is coordinated with the model yield mark by changing weights in the neural system.

LITERATURE SURVEY

Assignment scheduling calculation is a strategy by which undertakings are coordinated, or designated to server farm assets. Because of clashing scheduling destinations, for the most part, no completely consummate scheduling calculation exists. A decent scheduler actualizes an appropriate trade-off, or applies a mix of scheduling calculations as indicated by various applications. An issue can be comprehended in seconds, hours or even years relying upon the calculation connected. The productivity of a calculation is assessed by the measure of time important to execute it. The execution time of a calculation is expressed as a period multifaceted nature capacity relating the info. There are a few sorts of time unpredictability calculations that show up in the writing [2]. In the event that an issue has a polynomial time calculation, the issue is tractable, doable, effective or sufficiently quick to be executed on a computational machine. In computational intricacy hypothesis, a set of issues can be dealt with as multifaceted nature class taking into account a specific asset [2].

Class P is the arrangement of choice issues that are reasonable on a Deterministic Turing Machine in polynomial time, which implies that an issue of Class P can be chosen rapidly by a polynomial time calculation.

Class NP is the arrangement of choice issues that are resolvable on a Nondeterministic Turing Machine in polynomial time, yet an applicant arrangement of the issue of Class NP can be affirmed by a polynomial time calculation, which implies that the issue can be confirmed rapidly.

Class NP-complete is the arrangement of choice issues, to which all other NP issues can be polynomially transformable, and an NP-complete issue must be in class NP. As a rule, NP-complete issues are more troublesome than NP issues.

Class NP-hard is the arrangement of streamlining issues, to which all NP issues can be polynomially transformable, yet an NP-difficult issue is not as a matter of course in class NP.

Albeit the vast majority of NP-complete issues are computationally troublesome, some of them are tackled with worthy effectiveness. There are a few calculations, the running time of which is not just limited by the measure of the contribution of an illustration, additionally by the greatest number of the cases. Undertaking scheduling issue [3] is the issue of coordinating errands to various arrangements of assets which is formally communicated as a triple (T, S, O) where "T" is the arrangement of assignments, each of which is an occurrence of issue, the arrangement of doable arrangements is "S" and the goal of the issue is 'O'. Scheduling issue can be further arranged into two sorts as streamlining issue and choice issue in light of target O. An advancement issue requires finding the best arrangement among all the plausible arrangements in set S. Not the same as an improvement; the point of choice issue is generally simple. For a predefined doable arrangement $s \in S$, the issue needs a positive or negative response to whether the goal is accomplished. Obviously, enhancement issue is harder than the choice issue. Scheduling issues have a place with a wide class of combinational improvement issues going for finding an ideal coordinating of errands to various arrangements of assets. A simple issue alludes to one with a little number of the illustrations, so it can be basically worked out by polynomial calculations or identifications. In actuality, an issue is in Class NP-complete if its motivation is settling on a choice, and is in Class NP-hard if its motivation is an advancement.Scheduling problem categorization is as shown below in Figure 1.

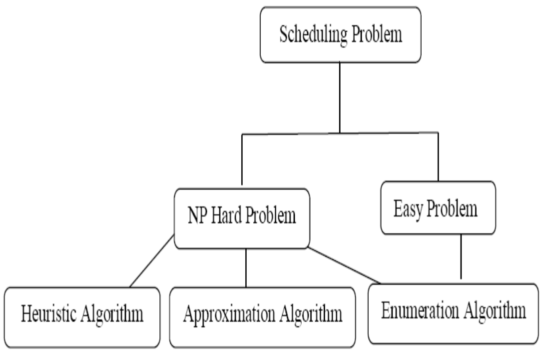

Figure 1: Scheduling Problem

Writing survey has been done in the region of Cloud figuring and certain other streamlining systems appropriate to the field of study. NavjotKaur et al (2010) dissected the correlation of work process Scheduling calculations in Cloud Computing. Calculations are contrasted and each other on the premise of parameters like aggregate execution time, the execution time for calculation, evaluated execution time.

Fatma A. Omara et al (2010) broke down a dynamic errand scheduling model which was utilized to enhance the fluffy choice in assignment scheduling on a system of handling components by acquainting new info parameters with a current fluffy model and, in the same time, enhancing the heap parity on the system in a dynamic situation. In this model, undertakings are produced arbitrarily and served in view of First-Come-First-Serve premise. The adjusted fluffy rationale model prompts more exact fluffy choices even while managing a bigger number of processors and/or bigger number of

assignments while expanding the quantity of included parameters in the fluffy model, with respect to a current one. Sih et al (1993) displayed aggregate time scheduling heuristic called Dynamic Level Scheduling (DLS), which represents bury processor correspondence overhead, when mapping priority compelled, conveying errands onto heterogeneous processor designs, with constrained or potentially sporadic interconnection structures. This strategy utilizes progressively changing needs to match errands with processors at every progression, and calendars over both spatial and worldly measurements to wipe out shared asset conflict.

This technique was quick, adaptable, broadly targetable, and shows promising execution. Additionally, another aggregate time scheduling system called Dynamic-Level Scheduling (DLS) was proposed and it represents bury processor correspondence overheads, when mapping priority charts onto numerous processor models. This system dispenses with shared asset dispute by performing scheduling and directing all the while to empower the scheduling of all interchanges and additionally all calculations. In heterogeneous handling situations, it represents shifting processor speeds and conveys a more watchful distribution of preparing assets. The calculation was part into two segments to allow a quick retargeting to any sought various processor engineering by stacking in the right topology-subordinate. The DLS procedure is quick, and iterative methodologies are intended to diminish the scheduling bottleneck which may demonstrate valuable. The association between scheduling and steering likewise justifies further examination. Since past correspondence asset reservations may obstruct a hub, from being planned on a specific processor, the rerouting of information exchange ways may encourage a superior hub processor mapping. SandeepTayal et al (2011) built up an improved calculation in view of the Fuzzy-Genetic Algorithm advancement which settles on a scheduling choice by assessing the whole gathering of the undertaking, in the occupation line. A two-level errand scheduling system in light of burden adjusting in distributed computing depicts this assignment scheduling component, which fulfills client necessities, as well as gives high asset usage. However, it needs more enhancements and this entire calculation depended on the precision of the anticipated execution time of every undertaking. Second, by, the productivity of the expectation utilizing Kernel Canonical Correlation Analysis (KCCA) strategy is exceedingly influenced by the decision of undertaking vector.

AlexandruIosup et al (2011) broke down the execution of distributed computing administrations for investigative registering workloads. It portrayed distributed computing administrations for Many-Task Computing and its applications traverse an expansive scope of conceivable setups, however using substantial quantities of figuring assets, over brief timeframes to fulfill numerous computational undertakings, where the essential measurements are in seconds. The reproduction results demonstrate that the present mists require a request of size in execution change, to be helpful to mainstream researchers, and shows which enhancements ought to be viewed as first to address this error amongst offer and request.

Rafael Moreno et al (2011) approved the difficulties and suitability of conveying a registering group on top of a multi-cloud framework spreading over four diverse locales for tackling approximately coupled MTC applications. The framework was investigated for the execution of various bunch designs, utilizing the group throughput (i.e., finished employments every second) as for execution metric. Distinctive group arrangements were thought about and the practicality of the Multi-Cloud arrangement was demonstrated from a cost viewpoint perspective. Shiyao Chen et al (2011) exemplified on time-changing asset use. A change is utilized to decrease the logged off issue with time differing processor limit with a steady limit. For web scheduling of under stacked framework, it is demonstrated that the Earliest Deadline First (EDF) scheduling calculation accomplished better aggressive proportion. For the over-burden framework, a web

scheduling calculation V-Dover is proposed with asymptotically ideal aggressive proportion when a specific acceptability condition holds. The outcome demonstrates that the proposed V-Dover calculation beats the best-known calculation in all cases contrasted and different norms.

Shuo Liu et al (2010) displayed a novel utility accumulation scheduling calculation for ongoing distributed computing administrations. The calculation that necessities to compensate the early culminations as well as to punish the premature births or due date misses of continuous errands. Calculation precisely picks the high master table assignments to execute, furthermore forcefully expels the undertakings that possibly prompt vast punishment. The execution of novel utility collection scheduling calculation was superior to the conventional scheduling calculations, for example, the Earliest Deadline First (EDF), the customary utility gathering scheduling calculation and an early scheduling approach in view of the comparative model. Jinhua Hu et al (2010) proposed a scheduling procedure on burden adjusting of Virtual Machine (VM) assets in view of hereditary calculation. This methodology processes the impact it will have on the framework ahead, after the arrangement of the required VM assets and afterward picks the minimum compelling arrangement, through which it accomplishes the best load adjusting and lessens or stays away from element movement. This technique takes care of the issue of burden unevenness and high movement cost by customary calculations subsequent to scheduling[12].

Table 1: Summary of Algorithms [11]

Method Used in Algorithm	Factor Considered	Advantages	Tool Used
DBD-CTO algorithm [4]	Cost, Time	It lowers the cost of computation and completes a task ingiven time boundary.	Java Environment
Improved Cost-Based TaskScheduling Algorithm [5]	Performance, Cost	It measures resource cost as well as computationalthe performance also improves (computation/communication) ratio.	Cloud Sim
A PSO-based Heuristic forScheduling WorkflowApplications [6]	Cost of computation,Cost of dataTransmission	It gives three times cost saving as compare to BRS andalso balances the load on resources by distributing tasksto available resources.	JSwarm package
Multi-Objective TaskAssignment in Cloud Computingby Particle Swarm Optimization[7]	Processing andTransferring time,Processing andTransferring cost	It is not only optimizes the time, but at the same timeoptimizes the cost also.	Matlab R2009b
Bi-Criteria Priority basedParticle Swarm Optimization[8]	Execution time andExecution cost	It minimizes the execution cost while meeting thebudget and deadline constraint.	Java Environment
Independent Task SchedulingBased on GA [9]	Consider resource and time utilization.	Consider resource and time utilization.	CloudSim
Genetic Simulated AnnealingAlgorithm [10]	QOS Parameters, Cost	Considers the QOS requirements of different user tasks.	Java Environment

The principle scheduling parameters considered in the already specified strategies are recorded underneath:

- **Makespan:** It is the aggregate fruition time of all assignments in an occupation line. A decent scheduling calculation dependably tries to decrease the makespan.

- **Deadline:** It is characterized as the timeframe from presenting an assignment to the time by which it must be finished. A decent scheduling calculation dependably tries to keep the assignments executed within the due date imperative.
- **Execution Time:** This is the accurate time taken to execute the given assignments. Minimize execution time is a definitive point of a decent scheduling calculation.
- **Completion Time:** Completion time is the time taken to finish the whole execution of work. It incorporates the execution time and defers brought about by the cloud framework. Minimizing fulfillment time of assignments is considered by a large number of the current scheduling calculations.
- **Energy Consumption:** Energy utilization in cloud server farms is a present issue that ought to be considered with more care nowadays. Numerous scheduling calculations were created for diminishing force utilization and enhancing execution and consequently making the cloud administrations green.
- **Performance:** Performance demonstrates the general productivity given by the scheduling calculation to give great administrations to the clients according to their necessities. A decent scheduling calculation ought to consider the execution at the client end and additionally the cloud administration supplier end.
- **Quality of Service:** Quality of administration incorporates numerous client information requirements like meeting execution cost, due date, execution, cost, makespan, and so on. All are characterized in SLAs which is an agreement report characterized between the cloud client and cloud administration supplier.
- **Load Balancing:** It is the technique for dispersion of the whole load in a cloud system crosswise over various hubs and connections so that at once no hubs and connections stay under stacked while a few hubs or connections are over-burden. A large portion of the scheduling calculations attempts to keep the heap adjusted in a cloud system so as to expand the productivity of the framework.

PROPOSED TECHNIQUE

This paper used the idea of a coordinated resource administration platform called Harmony, to this model two extra functionalities are added one is to figure the ideal time period and another is to limit the unauthorized access.

The proposed model gives collaboration between distinct public clouds and gives trust administration to accessing required files. The proposed model is depicted in Figure 2.

As the architecture represents that the harmony platform comprises of four major parts:

- Resource Management
- Trust Management
- Resource Selection
- Optimal time period calculation

The major part is the resource management where the service supplier or data proprietor will upload the data into the obliged or intrigued cloud server. The decision of the cloud server is chosen by the client.

Resource Management

Before uploading the data client has to create a VM with pre-described edge for data restrain in cost assisted manner.

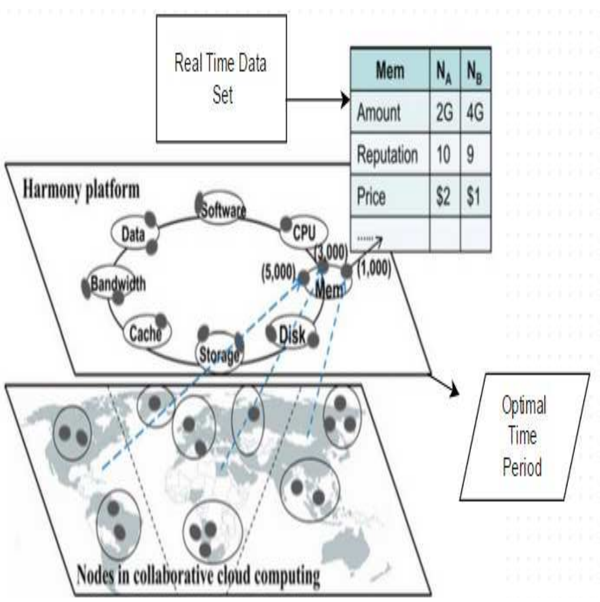

Figure 2: Proposed Model Architecture

This paper utilizes a trust manger as the base to check the client character and to maintain the collaboration between the clouds. Figure 3 portrays the stream of the Resource management

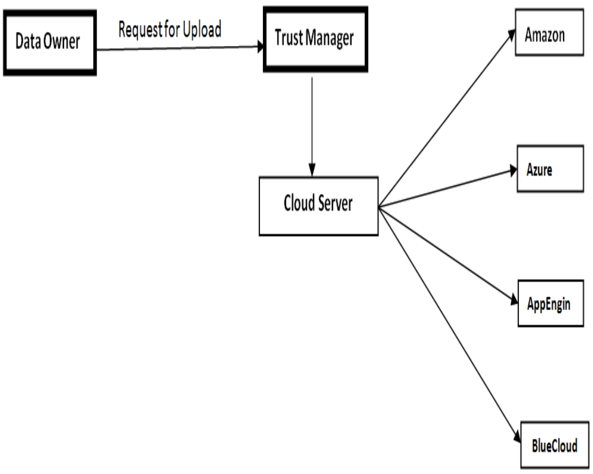

Figure 3: Resource Management Phase

The algorithm for trust management is as per the following:

- A user creates the VM in any one of 4 clouds with the required size and a threshold limit for storing the data.
- Then user logins to that cloud using the login credentials, if the credentials are not matched then a user will be rejected.

- Then the user uploads his data into the cloud if the threshold exceeds then upload will fail.
- Else file will be stored in the desired cloud.

Then the trust manager keeps the required information about the file and a trusted key is generated for each file and stored in three places:

- Trust manager
- Cloud server
- Data Owner

Then the resources are efficiently placed in the required cloud and the process of resource management ends with the successful completion of the Owner request.

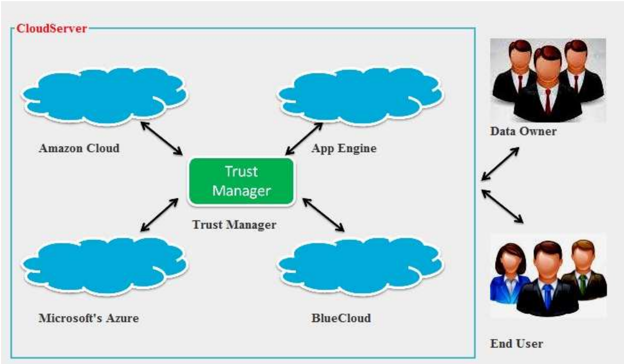

Figure 4: Resource Management Structure

The figure 4 portrays 4 virtual clouds created for resource sharing and the Trust Manager as the central base for the proprietor and client demands.

Trust Management

As the paper defines a trust-worthy resource sharing a trust management process is required to authenticate the user requests for the resources. The user will request the trust manager with his credentials for the proper resources. The trust management works as depicted in figure 5:

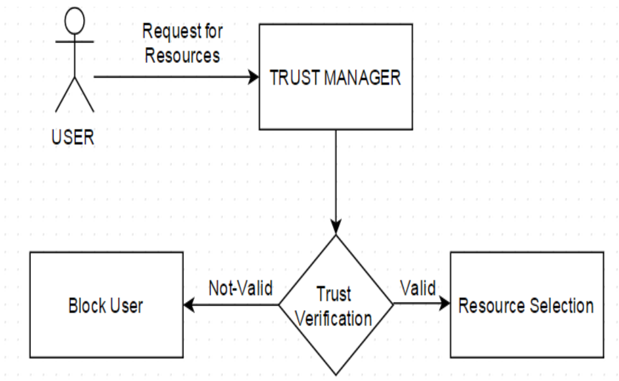

Figure 5: Trust Management

The user sends a request to the trust manager for the required resources along with the credentials and a secret key. The trust manager then verifies the trust composition of the user if the trust value is valid then the trust manager forwards the request to the resource selection else the user is blocked and notified.

Resource Selection

Resource selection is the procedure of discovering where the actual resource is located and acquiring that record utilizing cloud collaboration. The Neural Network (NN) model is utilized to choose the appropriate resources. The resource determination procedure utilizes a QoS.

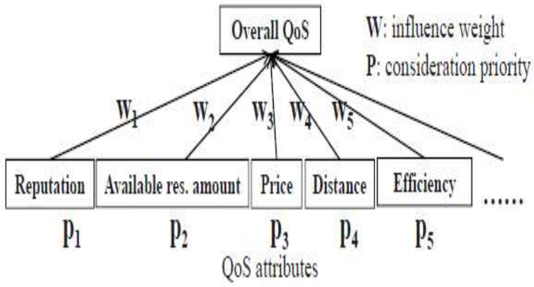

Figure 6: Resource Selection Using NN

Oriented Resource Selection for better cloud collaboration. The NN process searches all the clouds for the obliged resources and brings the conceivable one and gives to the client.

The NN procedure gets a set of parameters as input and those inputs are the clouds file index values and the values are given to the summation function and the activation function decides the winning value. The winning assessment defines that the required resource is located at that particular cloud. The process of NN is depicted in Figure 6.

As the depicts portrays that the {W1, W2,Wn} speaks to the weight value and the {P1, P2,Pn} characterizes the need values.

The input for the resource choice is the QoS parameters and the yield will be the resource location.

Optimal Time Period Calculation

The enhancement to the Harmony is to find out the optimal time period for resource selection. The NN process starts searching the required resources throughout the clouds using the cloud collaboration. The time required to search and fetch a request must be calculated to do so a triggering function is incorporated into the trust manager.

The triggering function is triggered when the process of searching starts and the stopped after receiving the resources back to the user. The optimal time period is calculated in Milliseconds.

ANALYSIS

The proposed model is tested under various resource management and resource selection conditions. The main metrics considered for the analysis are:

- Successful resource selection VS failures
- Waiting Time

The first measurement defines the total number of resource requests and the number of successful resource selection vs the failure requests due to delay.

The comparison is made with the Power Trust [9] algorithm. Figure 7 shows the successful resource selection for groups of 5-20 requests by each method. The black color represents delayed successful requests that have waited in the queue before being processed, and the grey color represents successful requests with no delay. As the graphs depict that the PowerTrust generates a large number of delayed successful requests, while the proposed methods generate no delayed requests.

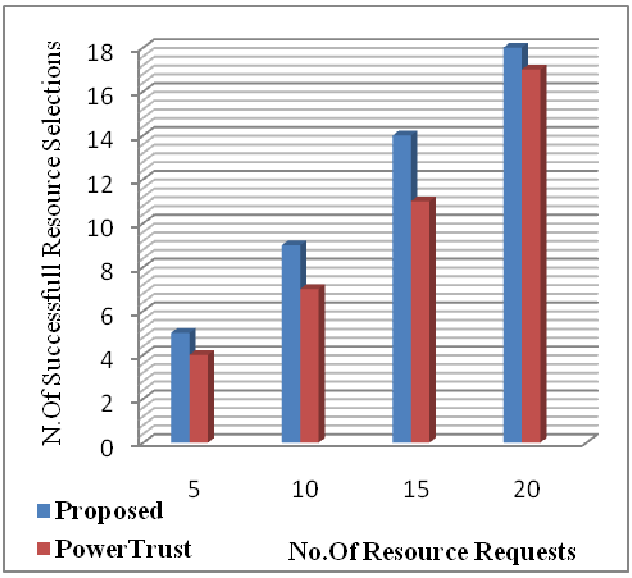

Figure 7: Successful Resource Selections

As the figure clearly depicts that the proposed system outperforms the PowerTrust with a minimum number of failures.

The next metric to analyze is the waiting time to complete a resource request. Again the proposed system is compared with the PowerTrust for efficient analysis.

Figure 8 shows the total waiting time for each group 5-20 requests, including failed requests. We see that PowerTrust generates high delay for a request, while the proposed method produces little or no delay, which is consistent irrespective of the number of requests. This is because PowerTrust always chooses the highest overall reputed nodes as resource providers without considering node load. These nodes receive too many requests, causing many to wait in the queues. Since proposed method select lightly loaded nodes as resource providers, they generate few delayed requests.

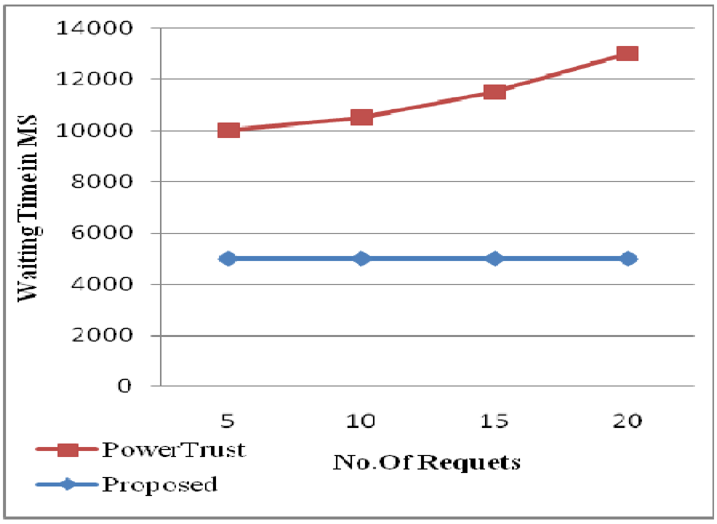

Figure 8: Waiting Time Delay

The figure clearly depicts that the time taken by the Power Trust is more compared to the proposed method.

CONCLUSIONS

This paper proposed an efficient resource selection algorithm called Harmony and added some functionality like an optimal time period and unauthorized request blocks for efficient trust management. The resource management and QoS resource selection provide efficient collaboration between the cloud users. The NN phase takes less waiting time to complete the requests with higher success rates. Further, the trust manager maintains all the sensitive information related to the resources and kept confidential among the cloud users.

REFERENCES

1. *D. Abramson, R. Buyya, and J. Giddy, A computational economy for grid computing and its implementation in the nimrod-g resource broker, Future Gener. Comput. Syst. 18 (2002), no. 8,1061{1074.*
2. *M. Sipser. Introduction to the Theory of Computation. International Thomson Publishing, 1st edition, 1996. 29*
3. *Saranya .S, Saranya. N," An Efficient Resource Allocation for Improving Resource Utilization in Self Organizing Clouds", International Journal of Innovative Research in Computer and Communication Engineering, Volume.2, Special Issue 1, March 2014*
4. *CristianMateos, ElinaPacini & Carlos GarcGarino, (2013), An ACO-inspired algorithm for minimizing weighted flowtime in cloud-based parameter sweep experiments.*
5. *BaominXu, Chunyan Zhao, Enzhao Hu, Bin Hu, (2011), Job scheduling algorithm based on Berger model in cloud environment, Advances in Engineering Software 42, PP. 419–425.*
6. *Kapadia, G. S., &Gulati, R. M. (2014). Security concern for virtualization in cloud computing. International Journal of Computer Science Engineering and Information Technology Research, 4(5), 107-116.*
7. *Wei Wang, GuosunZeng, Daizhong Tang and Jing Yao, (2012), Cloud-DLS: Dynamic trusted scheduling for Cloud computing, Expert Systems with Applications 39, PP.2321–2329.*
8. *Gang Liu, Jing Li and JianchaoXu, (2013), An Improved Min-Min Algorithm in Cloud Computing, Department of Computer Science and Engineering, Changchun University of Technology, China.*

9. *Suraj Pandey1, LinlinWu1, SiddeswaraMayura Guru2 &RajkumarBuyya, A Particle Swarm Optimization-based Heuristic for Scheduling Workflow Applications in Cloud Computing Environments, Cloud Computing and Distributed Systems Laboratory, Department of Computer Science and Software Engineering, The University of Melbourne, Australia.*
10. *Kaleeswaran, A., Ramasamy, V., &Vivekanandan, P. (2013). Host Scheduling Algorithm Using Genetic Algorithm In Cloud Computing Environment. International Journal of Advances in Engineering & Technology.*
11. *J. Kennedy and R. Eberhart, (1995), Particle swarms optimization In IEEE International Conference on Neural Networks, volume 4, pages 1942–1948.*
12. *Mridul S. Kuma et al., Cloud Computing Usability in Mobile Communication Network, Vol. 2(3), 105-110*
13. *ShamsollahGhanbaria& Mohamed Othmana, (2012), A Priority based Job Scheduling Algorithm in Cloud Computing, Procedia Engineering 50, and PP. 778 – 785.*
14. *PoojaSamal and Pranati Mishra, (2013), "Analysis of variants in Round Robin Algorithms for load balancing in Cloud Computing", International Journal of Computer Science and Information Technologies, pp. 416-419, Vol. 4 (3).*
15. *BrototiMondala, KousikDasgupta&ParamarthaDuttab, (2012), Load Balancing in Cloud Computing using Stochastic Hill Climbing-A Soft Computing Approach, Procedia Technology4,PP.783–789*